Automated Classification and Localization of Daily Deal Content from the Web

Automated Classification and Localization of Daily Deal Content from the Web

John Cuzzola, Jelena Jovanovic, Ebrahim Bagheri and Dragan Gasevic

Abstract

Websites offering daily deal offers have received widespread attention from the end-users. The objective of such Websites is to provide time limited discounts on goods and services in the hope of enticing more customers to purchase such goods or services. The success of daily deal Websites has given rise to meta-level daily deal aggregator services that collect daily deal information from across the Web. Due to some of the unique characteristics of daily deal Websites such as high update frequency, time sensitivity, and lack of coherent information representation, many deal aggregators rely on human intervention to identify and extract deal information. In this paper, we propose an approach where daily deal information is identified, classified and properly segmented and localized. Our approach is based on a semi-supervised method that uses sentence-level features of daily deal information on a given Web page. Our work offers *i)* a set of computationally inexpensive discriminative features that are able to effectively distinguish Web pages that contain daily deal information; *ii)* the construction and systematic evaluation of machine learning techniques based on these features to automatically classify daily deal Web pages; and *iii)* the development of an accurate segmentation algorithm that is able to localize and extract individual deals from within a complex Web page. We have extensively evaluated our approach from different perspectives, the results of which show notable performance.

1. Introduction

The World Wide Web has given rise to a digital marketplace where goods and services of all varieties are sold. This arena is no longer the domain of solely traditional brick and mortar retail outlets. Forrester research predicts, by 2017, Americans will spend $370 billion via e-commerce, up from $262 billion in 2013; it is also expected that e-commerce growth will outpace sales growth at bricks-and-mortar stores over the next five years [Mulpuru et al., 2013]. Perhaps the greatest indicator of this phenomenon is the emergence of deal collectors and deal aggregation services. Deal collector sites, such as GROUPON, have staffed 10,000 employees to locate special product offers that bargain hunters are on constant lookout for [Ghigliotty, 2011]. A plethora of such sites have lead to the creation of deal aggregators – sites that track bargains found by multiple deal collectors. Even Google, arguably the reigning king of search engines, has its own deal locator service known as Google Offers. However, Google's dominance in information retrieval has yet to extend to Google Offers which is still in its infancy with only a beta deployment to a handful of cities. This suggests there is still an opportunity to make a significant impact in this category of Web search.

For deal aggregator Websites to be successful, they need to effectively handle the challenge of identifying and indexing a large majority, if not all, of the daily deals that are offered in different

place across the Web. Unlike traditional e-commerce Websites such as Amazon and eBay, daily deal Websites have features that are unique such as:

1. *High frequency of change:* The content of daily deal Websites is updated very frequently, e.g., deals can be posted on a deal Website and expire only after a couple of hours. Therefore, the detection and dissemination of deal information is very time sensitive;
2. *Lack of content classification:* Daily deals are often posted on deal Websites in the form of a long list of items often without proper categorization. Unlike other e-commerce Websites that list categorically similar items besides each other, daily deal Websites often have other criteria for listing items such as the amount of time remaining before the expiry of a deal; therefore, items in a list could be quite heterogeneous;
3. *Scarcity of product or service description:* The deals that are listed on daily deal Websites often do not contain the complete description of the products or services that are offered. Typically, it is only an overview that is provided and the users are encouraged to visit the actual product or service Web page for more detailed information.

Given such characteristics, many deal aggregators rely highly on human intervention to identify and extract daily deals from the Web. It is upon employees of daily deal aggregators to manually or semi-manually browse through daily deal Websites and collect the information in a timely and accurate manner. Clearly, this is a time-consuming, expensive and tedious task. Our main objective in the work presented in this paper is to address this challenge in order to reduce deal aggregators' heavy reliance on human intervention for finding daily deals. With this research objective in mind, we propose intelligent deal crawler technology that is able to quickly and accurately crawl the Web, identify Web pages that contain deal information, and properly segment such Web pages into localized segments that contain only one individual daily deal. These localized daily deals are indexed as to contribute to the automated aggregation of a daily deal dataset. In particular, our work offers the following contributions:

1. The identification of effective and computationally inexpensive set of features that are able to discriminate between Web pages that consist of daily deal information and those that do not;
2. The construction and systematic comparison of different learning machines based on the identified features that can efficiently classify pages as containing daily deal information with emphasis on both recall and precision;
3. The development of a segmentation algorithm capable of determining the number of daily deals present on a given Web page, and accurately localizing those segments of the page that contains information about one particular deal.

The rest of the paper is organized as follows. Section 2 reviews background literature as it pertains to Web page classification and segmentation. Section 3 provides an overview of our proposed approach, which is followed by the description of the selected discriminant features and technical details of the proposed approach. We report on five different forms of extensive evaluation that we have performed on various aspects of our approach in Section 4. The paper is then concluded with some important points of discussion, future work and concluding remarks.

2. Background

The approach proposed in this paper, further referred to as the Deal Finder approach, consists of three main phases: 1) Web page classification into deal and no-deal pages (candidate selection), 2) segmentation of deal-pages (identifying regions of the page that do and do not contain products) and 3) localization of individual deals within the segmented page. Accordingly, in this section we give a concise overview of the previous research work in the areas of Web page classification (Section 2.1) and Web page segmentation (Section 2.2) in order to properly position our work with respect to the existing knowledge and work in the field.

2.1 Web Page Classification

Web page classification can be defined as a supervised learning task of assigning a Web page to one or more of predefined categories. These categories can be related, for instance, to the topic of the page (e.g., politics, culture, sport), or function of the page (e.g., homepage, deal page), or the presence of spam or pornographic content on the page. Based on the number of categories used in a particular classification task, one can distinguish between binary and multiclass classification tasks. A distinction is also made between single-label and multi-label classification, where the former means that only one class can be assigned to an instance, while the latter allows for assigning more than one class to each instance. Finally, classification can be divided into hard or strict (an instance is definitively assigned to one of the classes) and soft (an instance is assigned a probability of being a member of each of the considered classes). There are also other ways of categorizing the classification task, but these four are of relevance for the problem/task we deal with. In particular, our task of classifying Web pages into deal vs. no-deal pages can be characterized as functional, binary, single-label and strict classification.

Qi and Davidson [2009] have provided an excellent review of the features and algorithms used for performing the classification tasks, with a special emphasis on the utilization and benefits of Web-specific features and methods. As for features, they made a broad distinction between on-page features and features of neighbours; the former include features located on the page to be classified, whereas the latter comprise features of pages that are in some way related to the page to be classified (e.g., parent, child, neighbour). For instance, Chau & Chen [2008] proposed a method for binary, single-label Web page classification that relies on a combination of on-site features (the content and URL of the Web page) and neighbour features (the content of the page's neighbours). These features served as inputs to the Neural Network (NN) and Support Vector Machine (SVM) classifiers these authors experimented with. The NN classifier proved to be significantly better than SVM both in terms of accuracy and F-measure. In the work presented in [Selamat et al., 2003], a neural net was used for a multiple category Web Page classifier for newspaper sports articles trained on the most frequently occurring words. Fiol-Roig et al. [2011] experimented with multiple variants of decision trees and recorded a 91% (±1%) Web page classification accuracy amongst the variants.

The approach proposed in this paper is based on the on-page features, thus we further consider only this category of features. Most often on-site features are text-based features, that is, features that can be derived from the text of a Web page. Typical features of this type include: bag-of-words and/or n-gram representation of the page content, summary of the page content, HTML tags, and URL of the page. For instance, Ozel [2011] proposed a Genetic Algorithm (GA)-based Web page classifier which uses as its input a set of features comprised of all the terms from the page and their corresponding HTML tags. Since the created feature set tends to be overly large, terms from similar HTML tags (e.g., , , <b> and <i>) are grouped to reduce the number of features. Furthermore, different weights are assigned to each feature and these weights are determined by the GA. An interesting set of features, partly text-based, was proposed by Selamat & Omatu [2004] as an input for a NN-based Web page classifier. To tackle the problem of classifying news articles into several different sport-related categories, they combine two different kinds of features: 1) features obtained by applying Principal Component Analysis (PCA) to a Web page represented by the term-frequency-weighting scheme; and 2) the class-profile features which are, in fact, manually selected, most frequent words in each class weighted using an entropy-weighting scheme. The proposed classification method proved particularly effective (in terms of classification accuracy) for classes that are represented with a small number of documents in the dataset. Chen & Hsieh [2006] also rely on the text of a Web page for the creation of two feature sets: 1) semantic features obtainable by applying Latent Semantic Analysis to the page text; and 2) 'regular' text features (e.g., the number of words and keywords, and the keywords-to-words ratio). Each feature set serves as input to an SVM-based classifier, and a voting policy is defined to determine the final classification of a page based on the results of the two classifiers.

Text-based features are sometimes combined with other kinds of on-site features, such as visual features, i.e., features derived from the visual representation of a page (as rendered by a Web browser). For instance, Ahmadi et al. [2011] makes use of three kinds of on-site features to develop a classifier capable of detecting pornographic Web pages. In particular, besides a set of text-based features (e.g., number of words, keywords and black-words), they also use a set of features comprising information about the page structure (links, images, videos, tooltips and the like), as well as a set of features derived from images on the page (specifically, topological and shape-based characteristics extracted from the skin region of images). These feature sets serve as input to three classifiers (one feature set per classifier) that are combined in a hierarchical structure to obtain the final robust classifier.

Our focus in the presented work is on text-based features. Namely, we focus on the bag-of-words representation of the page, and a set of features derived through part of speech tagging and name entity recognition over the textual content of the page as described in section 3.1.

2.2 Web Page Segmentation

The task of Web page segmentation has been approached with methods that could be classified into: template-based, vision-based, and tag-based [Kang et al., 2010]. As their name suggest, template-based methods require the creation of templates that would be used for the segmentation of a page into different content blocks. Initial methods assumed manual creation

of templates that typically comprise regular expression rules. However, as the task of template creation is laborious and needs to be repeated for each domain, novel approaches tend to automate it through the use of Machine Learning (ML) techniques. However, these approaches require a considerable number of Web pages for training the ML algorithm and building a template, so they also have not completely solved the problem. Another deficiency of template-based methods is that they rely on the DOM (Document Object Model) structure of Web pages. This allows for segmenting a page based on the structural information, i.e., on the structure of the page's DOM tree, but not on the content of individual DOM nodes.

Vision-based segmentation methods try to overcome these challenges through heuristic rules based on the visual or layout information of a Web page (after page rendering) such as font size and color, and distance between paragraphs. These methods rely on the assumption that similar content blocks are located close to one another and are visually similar. A well known work in this category is the VIPS (Vision-based Page Segmentation) algorithm developed by Microsoft researchers Cai et al. [2003]. However, as both the structure and visual appearance of Web pages tend to change rather often, rules that rely on them need to be frequently maintained which reduces the efficiency and appeal of the vision-based methods.

Tag-based methods rely on a predefined set of content-bearing tags and identify content blocks by measuring distance between such tags. For instance, an early work that followed this approach relied on the use of the <table> tag as the indicator of a content block [Lin & Ho, 2002]. However, as this way of structuring Web pages became obsolete, more recent work rely on other HTML tags (e.g., <p>, <ul>, <hr>) for segmentation of Web pages into content blocks (e.g., [Debnath et al., 2005]).

More recently, trying to overcome the observed deficiencies of the three aforementioned categories of segmentation methods, the research community came up with some completely novel approaches. For example, Kohlschütter and Nejdl [2008] approach the problem of Web page segmentation from a quantitative linguistic point of view, that is, as a problem of identifying significant changes of certain statistical properties within the textual content of a Web page. In particular, they experimented with token-level text density (the number of tokens in a text fragment) and demonstrated that their Block Fusion algorithm for Web page segmentation, that relies on this statistical feature of the text performs significantly better than the state-of-the-art graph-theoretic algorithm [Chakrabarti et al., 2008].

Another interesting approach to Web page segmentation is given in [Vineel, 2009]. The main idea is to characterize each node of the DOM tree structure of a Web page with two features: content size and entropy. The former feature indicates the amount of textual content contained in a node (and its child nodes), whereas the latter measures the presence of local patterns within the node's subtree (the more repetitive the patterns are, the higher the entropy of the respective node). These features of a page's DOM tree are used as an input to an unsupervised algorithm that automatically identifies segments of the given Web page.

Our approach is most closely related to the work of Kang et al. [2010] whose method of Web page segmentation is based on recognizing repetitive tag patterns in the DOM tree structure of a page. This method is rooted in the observation that well structured and repetitive layouts are

often used in Web pages to provide end users with consistent information. The proposed algorithm detects repetitive HTML tag patterns (named key patterns) in a Web page, and generates virtual nodes which correspond to page blocks, that is, primary units of page segmentation.

3. Proposed Method

Our method for the automation of identifying and localizing deal-specific information from the natural language text within a Web page is illustrated in Figure 1. We outline the tasks of our Deal Crawler as follows:

(a) A stack of candidate Web page URLs that may contain information on our target domain (deals) is obtained from a user or through automated Web-searches of known products. This preliminary step “seeds” the crawler for its starting point(s) in the analysis process.

(b) If stack is not empty then a URL is popped off the stack, and its HTML content is retrieved by a HTTP client.

(c) A trained binary classifier determines whether the text of the Web page contains deal information or not using our proposed classifier (Section 3.1). If the page is classified as not containing deals then it is discarded and the process repeats at step (b).

(d) Those pages classified as deal undergo page segmentation resulting in several segments per page (Section 3.2).

(e) Each of the extracted segments will in turn be recursively classified in an effort to localize individual products within each segment. This process is detailed in Section 3.3.

(f) Identified individual deals(products) are stored for later use for various applications (personal shopping assistants, product-specific domain searches, product data warehousing/trend analysis, and so forth).

(g) The deal page is then searched for URLs to other pages, and the identified URLs are added to the stack of candidate pages for consideration. The stack is sorted in descending order using a calculated *deal-class-membership* ratio thus placing those pages of higher confidence (most likely deals) towards the top (Section 4.2). Pipeline repeats at step (b) until the candidates stack is empty.

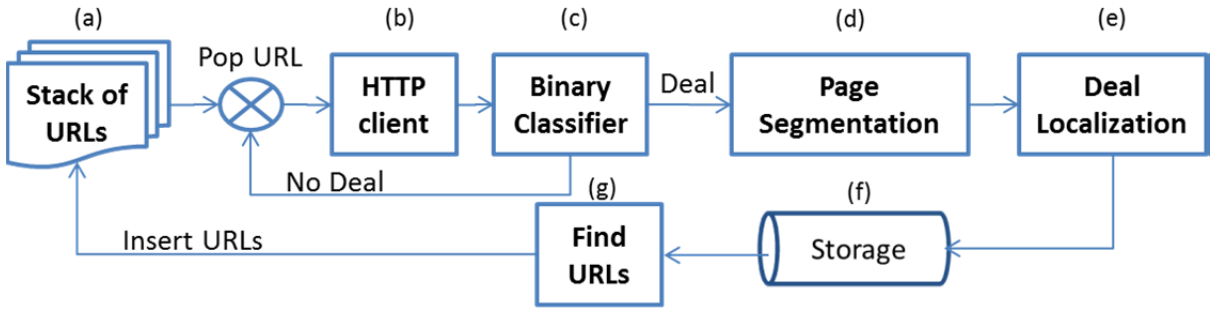


Figure 1. The pipeline for a Deal Crawler: (a) Initial seed URLs (b) HTTP client (c) Deal Classifier (d) Page Segmentation (e) Localization (f) Storage (g) Next candidate URLs

The core of our crawler resides with *binary classification* (c), *page segmentation* (d) and *deal localization* (e). In the proceeding sections we examine each of these components in detail.

3.1 The Deal Classifier (Binary Classification)

The deal classifier performs a binary(true/false) decision on whether a candidate Web Page likely contains product information (and subsequent localization is warranted), or the page should be discarded. This is achieved through *sentence level classification* where each individual sentence is evaluated as a *deal* or *non-deal* sentence. Specifically, we ask whether the language of the sentence is consistent with what would be expected within a product (deal) Web Page. Table 1 demonstrates examples of deal versus non-deal sentences including a probability that such sentences are from a deal Web Page as computed by the deal classifier.

Table 1. Two example sentences with probability of being in a deal Web page.

SENTENCES	DEAL PROBABILITY [0, 1.0]
Buy unlimited vouchers as a gift Package. Includes a 7" Google Android 2.3 Tablet with a 30 pin USB switch adaptor , charger and user manual Lightweight and easy to use. Perfect idea for people on the go. Makes a great gift !	0.9998
Challenges address the conceptualization how e-business related knowledge is captured , represented , shared, and processed by humans and intelligent software.	0.0248

To perform sentence level classification, we first extract features from the input sentences then input those metrics into a combined Naive Bayes (NB) text classifier and Expectation-Maximization (EM) clustering calculation. The features are summarized in Table 2. As indicated in the last column of the table, we make use of lexical knowledge bases, such as WordNet[1], and natural language processing techniques to obtain features from the plain text of a Web page.

The Words feature (A) includes individual occurrence of frequently appearing words as these are common to almost any text classification task. In addition, we use the lexical knowledge base WordNet to include words with similar meanings (synonyms) or other closely related words, in order to increase the likelihood of exposure to words that may appear in the wild but were not seen during the training. We also utilize named entity recognition to capture semantics

[1] http://wordnet.princeton.edu/

of words. This resulted in an additional seven features (B-H) such as currency, percentage and organization entities, among others. Part of speech tagging is also used to include features that identify simple counts such as average dollar value of a block of sentences (I) and number of symbols (such as punctuation marks) in the block (L). Named entity recognition and part of speech detection for features (B-H) was provided through the OpenNLP framework (http://opennlp.apache.org/).

Table 2. Features extracted for training and testing (ALL).

Reference	Name	Description	Source
A	Words	Frequently appearing words in the sentence block and their synonyms and other related words obtained via WordNet.	WORDNET
B	ner_dateI	Number of dates identified through Named Entity Recognition (NER)	Named Entity Recognition (NER)
C	ner_organizationI	Number of organization instances identified through NER	
D	ner_timeI	Number of time-related instances identified through NER	
E	ner_locationI	Number of location instances identified through NER	
F	ner_percentageI	Number of percentages identified through NER	
G	ner_moneyI	Number of money values identified through NER	
H	ner_personI	Number of person instances identified through NER	
I	sym_dollarAvgI	The average dollar value identified through POS tagging.	Part Of Speech (POS) Tagging
J	sym_percentAvgI	The average percentage identified through POS tagging	
K	sym_CD_posI	Count of numerical values identified through POS tagging	
L	sym_SYM_posI	Count of symbols identified through POS tagging	

Utilizing these features, a set of cluster groups are found with Expectation/Maximization (EM) to identify deal versus non-deal characteristics within each group. The set of input sentences necessary for supervised training of the EM were comprised of both positive (deal) and negative (no-deal) examples. Positive training samples came from our industrial partner, a deal aggregator, which pools thousands of products from hundreds of deal and product sites on a daily basis. Negative training examples were obtained through texts that are either available under the Creative Commons license or are in the public domain and are accessible through repositories such as Project Guttenberg (http://www.gutenberg.org/). Table 3 gives an example of such negative sentences along with probability computed by the deal classifier.

Table 3. 'Negative' sentences taken from the text "*Marlborough and the War of the Spanish Succession, 1955*." with deal probability.

NEGATIVE (non-deal) SENTENCE	DEAL PROBABILITY [0, 1.0]
On reaching The Hague in July Marlborough at once entered into	0.020

negotiations for a last-minute settlement with France and Spain.	
That having failed, he began the reconstruction of a Grand Alliance against Louis XIV.	0.024
King William remained in the background, wisely leaving the Earl to treat with the ambassadors from the various courts of Europe.	0.086

Together, these positive and negative sources created a training corpus of 1.6 million sentence vectors[2], where each tuple of the vector υ = {A, B, … , L} represents a feature of Table 2.

Once this EM model is learnt, a probability can be computed for a block of sentences using a Naïve Bayes (NB) classifier. A sentence block is a sequential grouping of one or more sentences such as a paragraph of text. The NB classifier calculates the probability that a sentence block belongs to each EM cluster, and then a weighted average across all clusters completes the calculation. Formally, given n-clusters (C_n), discovered through EM learning, and f features (F_f) of the sentence block S, the probability of S belonging to a cluster C_i , denoted $P(C_i|S)$, using Naïve Bayes is:

$$P(C_i \mid S) = \alpha P(C_i) \prod_{1..f}^{j} P(F_j \mid C_i) \qquad (1)$$

where $\boldsymbol{\alpha}$ is the normalization constant.

A sentence block is classified as consistent with containing deal-like content if the sum of the likelihood of being a deal within all EM clusters exceeds a set threshold τ:

$$\sum_{1..n}^{i} P(C_i \mid S) P(F_{deal} \mid C_i) > \tau \qquad (2)$$

We have set this threshold (τ) to 0.90, and later provide empirical evidence for the chosen value (Section 4.5). Tables 1 and 3 show examples of this calculation.

The ratio of deal sentence blocks to no-deal sentences is compared to a threshold value (Γ) to determine the final deal/no-deal classification of a Web page. Formally, let Q be the set containing a web-page's sentence blocks. Let R be the subset of sentence blocks within Q that satisfies equation 2. Then, a page is classified as a deal page if the following condition is met:

$$\lim_{B \to |Q \setminus R|} \frac{|R|}{B} \geq \Gamma \qquad (3)$$

We also include a sanity check where if the Web page meets this threshold but does not contain any monetary artifacts, as determined by named entity recognition, then the Web page must be

[2] The corpus can be downloaded for replication studies from http://ls3.rnet.ryerson.ca/corpus1/

classified as ‘no-deal’. This is to filter out those sites that describe a product, but are not selling the product. Examples of this are vacation blogs that describe seasonal travel packages and product review pages. By disabling this sanity check, the classifier can be extended to identify general product pages whether or not the products are for purchase.

3.2 Web Page Segmentation

Web page segmentation is the process of partitioning a Web page into logically grouped sections either visually, structurally, or semantically to form cohesive objects [Kang et al., 2010]. It is a highly active area of research with numerous approaches: machine learning algorithms, text-based algorithms, pattern matching algorithms, graph-based algorithms, and rule-based algorithms [Yesilada, 2011].

In our work, we segment Web pages based on the structure of the Web page defined by HTML tags and by the semantic textual clues obtained through natural language processing. Our cohesive object is the localization of individual product offerings with supporting information such as description, price, and duration.

After the binary deal classifier deems a Web page as containing deals, the page is iteratively partitioned into smaller blocks by dividing its content based on the HTML structure, starting at the most general (outermost) block level. The textual elements within this block are combined and submitted to the classifier and evaluated in the same manner as described in Section 3.1. However, rather than evaluating the text of the entire page, only the text within this candidate segment is considered. If this block is classified as deal, it is split into smaller segments and in turn each segment is recursively examined by the classifier for deals.

The boundary for division is determined based on the longest sequence of HTML tags of the greatest frequency. For example, consider the sequence of HTML tags: <div class>,<div style>,<hr>,<div class>,<div style>
. The longest frequent pattern would be <div class>,<div style> occurring twice in the sequence. This partitioning scheme is similar to the work of [Kang et al., 2010], and we extend the concept by including patterns that involve both the html tag and tag attributes (<div class>) instead of only the HTML tag ( ). This allows for more varied frequent patterns and results in a better narrowing of the desired deal sections of the Web page. The process repeats iteratively for each newly segmented block until either the block is classified as containing no deals, or a frequent pattern of open HTML tags cannot be found. A sanity check that involves part-of-speech tagging ensures there is a minimum amount of semantic information preserved in each block and reduces the chance of over segmentation. The result of this process is a segmentation tree where each leaf node represents either a deal or no-deal segment of the Web page. Our algorithm is outlined in Algorithm 1 and described below.

We initialize the algorithm with a set (C) that contains the starting partitioning of the Web page at the coarsest level of granularity. Hence, this set begins with a single partition - the entire Web page (1.1). Each block in this set is classified as either deal or no-deal using the Naïve Bayes classifier described in Section 3.1 (2.2). Blocks labeled as no-deal are ignored with no further segmenting since they do not contain product offers of interest (deals). For those blocks

classified as deal, a frequent pattern is determined using the technique described above where the block is split into multiple segments along this pattern and added into our current block set C (2.4). Segmentation is complete when there are no further blocks in set C to classify and split (2).

Input: The HTML content of a Web page to be segmented

Output: The input page partitioned into deal and no-deal regions

Algorithm:

```
1. Let C be a set of candidate blocks of a Web page.
   1.1 Initialize C with the outermost block. Typically C←<HTML>…</HTML>
2. WHILE C is non-empty DO
   2.1 Let N = {}  (empty set)
   2.2 FOR BLOCK in C DO
       2.1.1 classify BLOCK as deal or no-deal.
       2.1.2 IF BLOCK is deal THEN
             2.1.2.1  add BLOCK to set N.   N ← N + BLOCK
       2.1.3 END IF
   2.3 END FOR      /* NEXT BLOCK */
   2.4 FOR ƒ in N DO
       2.4.1 Find the longest frequent pattern (LFP) HTML sequence of block ƒ.
       2.4.2 IF (LFP exists) THEN
             2.4.2.1 Let β be the set of blocks split along the LFP sequence
             2.4.2.2 Add β to set C. C ← C + β
      2.4.3 END IF
   2.5 END FOR     /* NEXT ƒ */
3 END WHILE
```

Algorithm 1: The Segmentation/Localization algorithm

3.3 Deal Localization

Our objective is to semi-automate the process of identifying deal-like Web sites and individual product offerings for ultimate inclusion in a searchable database. The visual appearance and structure of a candidate Web page varies greatly. Furthermore, the degree of information available for a product also varies substantially with some sites offering supplementary descriptions, savings, discounts, value, and expiry data while others give only the mandatory product name and price fields. Consequently, determining the optimal stopping point in the segmentation tree is challenging; thus we have established a post-partitioning algorithm to further refine the output.

The most favorable partitioning is a block that contains a single product with price and name information and as many applicable supplementary fields available. Accordingly, we have devised heuristic rules to help enforce the desired characteristics; namely that each localized area: 1) contains a single product and 2) contains at least minimally sufficient information of product name and price.

While the initial segmentation uses a top-down approach, this refinement involves a bottom-up strategy. The initial top-down segmentation is particularly useful in discarding large sections of the Web page that are irrelevant (no deals) such as banners, headers, footers, and navigation sections. The bottom-up algorithm attempts to identify over partitioning by pruning child nodes in favor of their parent. This is accomplished through a series of rules derived through observation of numerous segmentation parse trees. In this section, we propose three such rules alongside examples to demonstrate our motivation and rationale.

Rule 1: A deal parent that produces only leaf nodes becomes a leaf node if and only if all its children are non-deal nodes.

To understand how this rule was derived consider the sentence block: "*Private pilot starter package with ground school, text books and school bag, and discovery flight C$595 list price - 55% off – Save C$326*". Figure 2 shows how our segmentation algorithm partitioned this text.

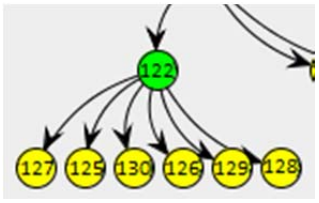
{122}={ **{125}**={Private pilot starter package with ground school, text books and school bag, and discovery flight} **{126}**={C$595 list price} **{127}**={-} **{128}**={55% off} **{129}**={– Save} **{130}**={C$326}}	

Figure 2: A tree with no deal children. Curly brackets in the text (left hand side) denote tree nodes. Green color denotes deal nodes, whereas yellow is used to indicate non-deal nodes.

The original full text (node 122) was partitioned into six children in which none were considered worthy of the deal label. Therefore Rule 1 is applied resulting in making node 122 a leaf of the tree. Hence, this rule prevents a deal node from being overly segmented to a state where none of its children retains the deal classification of its parent.

Rule 2: If a deal parent produces only leaf nodes and only one of its children is a deal node, then the parent becomes the leaf node.

In order to understand the reasoning behind this rule, first consider Figure 3 where this rule does not apply.

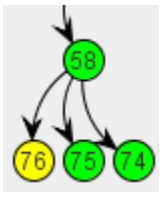
{58}={**{74}**={10% Off Special Nk'Mip Desert Cultural Centre Osoyoos Book with us and receive 10% off regular rates. View Deal} **{75}**={10% Off General Admission Squamish Lil'wat Cultural Centre Whistler Book tickets online save! Experience Whistler's Aboriginal heritage at this world-class attraction. View Deal} **{76}**={SEE ALL}}	

Figure 3: A tree with two deal children.

Node 58 contains two individual product offerings of "Nk'Mip Desert Cultural Centre" and "Lil'wat Cultural Centre" which were correctly partitioned into child nodes 74 and 75, with non-deal node 76. This divide and conquer approach is intuitive: if a node is classified as deal, then it

potentially contains more than a single product offering. Consequently, if this is the case, there must be two or more offspring classified as deal nodes.

Conversely, consider Figure 4 containing a node (121) with only a single product offering. This too produced three offspring but only one of which was a deal (node 131). Hence, there was no increase in the number of deals from the parent to the children. The rationale for this rule is if partitioning did not produce two or more deal nodes then the parent most likely only contains one individual deal. The offspring classified as non-deals are most likely supplementary information to the deal represented by the parent.

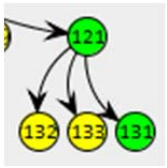
{121}={**{131}**={30-minute discovery flight and lesson C$200 list price - 51% off - Save C$101} **{132}**={Over 40 Bought} **{133}**={C$ 99}}	

Figure 4: A parse tree with one deal child; represents the sentence block “30-minute discovery flight and lesson C$200 list price - 51% off - Save C$101 Over 40 Bought C$ 99”

Rule 3: A leaf node that is classified as a deal but was prevented from further segmentation due to the part-of-speech sanity checking is treated as a non-deal node and subjected to Rules 1 and 2.

Figure 5 illustrates when this rule is useful. In this instance, the partitioning has produced deal children 209 (“Save $1 on General Admission”) and 211 (“Book online for $1 off General Admission”). However, although these nodes are classified as deal, the part-of-speech sanity checking has determined that there is insufficient text for further recursive segmentation. Consequently, the children are likely to have been over segmented and that its parent may be a better choice as a leaf node to the tree. Hence, nodes 209 and 211 are demoted to non-deal and rules 1 and 2 are then applied resulting in node 87 becoming a leaf node.

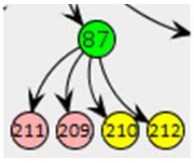
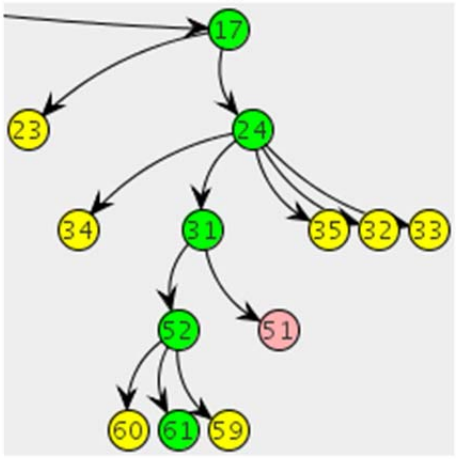
{87}={**{209}**={Save $1 on General Admission} **{210}**={Vancouver Aquarium Vancouver} **{211}**={Book online for $1 off General Admission} **{212}**={View Deal}}	

Figure 5: A tree with two deal children (209 & 211) flagged with few part-of-speech tags (made visually distinct by the use of the red color).

Figure 6 demonstrates the application of these rules to a branch of a tree. As seen in Part (a) of the figure, Node 52 becomes the leaf node due to Rule 2. In Part (b) of the figure, both leaf nodes 51 and 52 were classified as deal nodes, but node 51 was identified during the segmentation process as having too little semantic information thus its status remains as a deal node but it is treated as a non-deal candidate (Rule 3). Consequently, the parent node 31 becomes a leaf node after applying Rule 2 shown in Part (c) of the figure. The children of node 24 contain only one deal node (31). Hence, node 24 becomes a leaf node due to Rule 2, shown

in Part (d) of the figure. Since node 24 is now the sole deal node amongst its siblings (node 23), the tree is pruned to the parent node 17 depicted in Part (e). The process continues until no further pruning can be done on the leaf nodes resulting in the final segmentation tree as seen in Figure 6 Part (f). Figure 7 shows the full original unparsed segmentation tree and the portion of the tree that underwent the pruning process depicted in Figure 6.

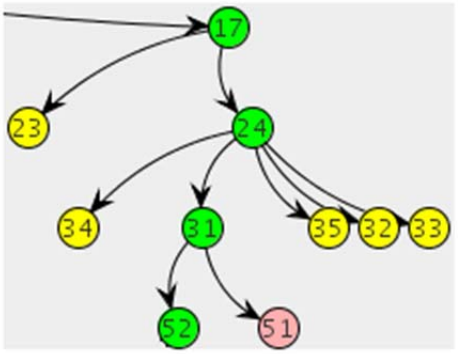
(a)



RULE 2

(59) 544 Users Alerted
(60) Nine West
(61) FINAL CLEARANCE at NINE WEST! Take an EXTRA 50% OFF the lowest ticketed price on all boots and booties - select styles may apply. See in store for details.

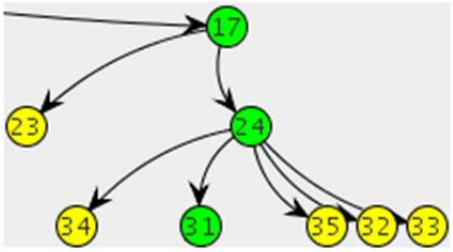
(b)



RULE 2 & 3

(51) 50%
(52) 544 Users Alerted Nine West FINAL CLEARANCE at NINE WEST! Take an EXTRA 50% OFF the lowest ticketed price on all boots and booties - select styles may apply. See in store for details.

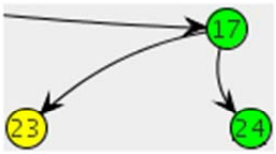
(c)



RULE 2

(31) 50% 544 Users Alerted Nine West FINAL CLEARANCE at NINE WEST! Take an EXTRA 50% OFF the lowest ticketed price on all boots and booties - select styles may apply. See in store for details.
(32) 14 location(s) Map It! ...
(33) Bayview Village Shopping Centre Map It!
(34) Dixie Outlet Mall Map It! ...
(35) Bramalea City Centre Map It! ...

(d)



RULE 2

(23) Live
(24) 50% 544 Users Alerted Nine West FINAL CLEARANCE at NINE WEST! Take an EXTRA 50% OFF the lowest ticketed price on all boots and booties - select styles may apply. See in store for details.14 location(s) Map It! ... Bayview Village Shopping Centre Map It Dixie Outlet Mall Map It! ... Bramalea City Centre Map It! ...

(e)

(17) Live 50% 544 Users Alerted Nine West FINAL CLEARANCE at NINE WEST! Take an EXTRA 50% OFF the lowest ticketed price on all boots and booties - select styles may apply. See in store for details.14 location(s) Map It! ... Bayview Village Shopping Centre Map It Dixie Outlet Mall Map It! ... Bramalea City Centre Map It! ...

(f)

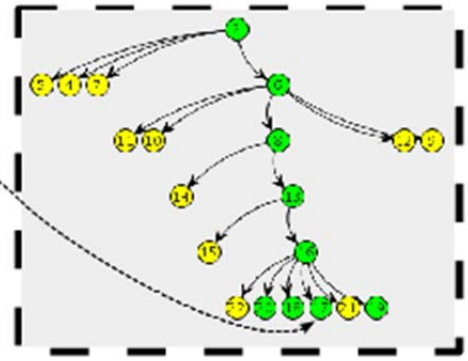


Figure 6. The tree pruning algorithm illustrated (green=deal, yellow=non-deals, pink=deal nodes but segmentation stopped by sanity checking)

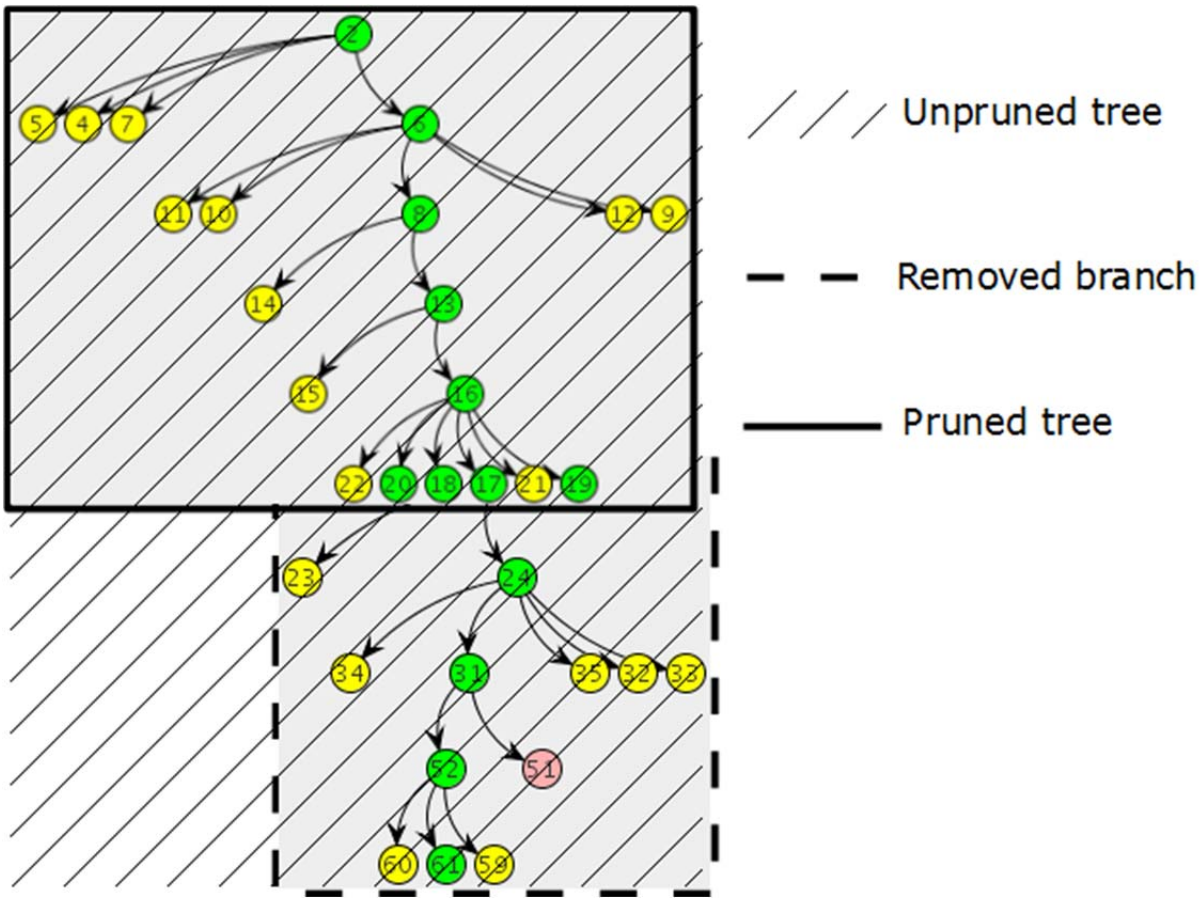


Figure 7: An unparsed segmentation (dashed area) that underwent the pruning process illustrated on Figure 6 (enclosed by the dashed line) to produce the final pruned tree (enclosed by the solid line).

4. Experiments and Evaluation

In this section we turn to the experiments we have done in order to evaluate the main phases of our proposed Deal Finder approach: 1) identifying a Web page as a potential deal candidate, and 2) localizing individual products within the candidate page. We also explore the ambiguity a deal crawler might encounter when performing general Web searches on its quest to finding worthy candidates for consideration (phase 1) and the difficulty it might have in pin-pointing specific products within deal pages (phase 2) .

In this section:

1. We examine the effectiveness of the combined Naive Bayes and Expectation / Maximization (NB/EM) classifier on the classification accuracy at the sentence level against three other common machine learning methods including Support Vector Machine (SVM), Neural Network (NN), and Decision Tree (J48).

2. We assess the classification accuracy at the page level. More precisely, these experiments were aimed at evaluating our method of Web page classification using the true-to-false ratio (T/F) of labeled sentence blocks.
3. We evaluate the page segmentation and deal localization algorithms by applying the sentence-level classifier recursively within an area of a Web Page.
4. We explore the impact of a reduced feature set on classification accuracy and propose an algorithm to perform feature reduction.
5. We justify our choice of threshold parameters for sentence-level classification and deal/no-deal region determination.

4.1 Evaluation of the Sentence-level Classifier

The core of our method for Web Page classification, segmentation, and deal localization depends on the ability of the sentence-level classifier to accurately label sentences as deal/no-deal. To evaluate the effectiveness of the Naive Bayes (NB)/Expectation-Maximization (EM) model, we compared its classification accuracy at the sentence block level with three other popular machine learning models: support vector machine (SVM), decision tree (J48), and neural networks (NN). Together these four models comprise a broad range of general machine learning categories: probability (NB/EM), optimization problem (SVM), graph model (J48) and activation function (NN). In addition, two different vector normalization techniques were investigated for training: linear scaling, and z-score normalization. To further improve the NB/EM classification accuracy, we employed the technique of ensemble voting [Dietterich, 2000]. This method involves training multiple independent classifiers with different, but perhaps overlapping, training sets. Each classifier provides its own probability calculation to individually determine deal or no-deal. The final class label is achieved by majority vote of the participating classifiers thus improving the overall accuracy by consensus. We trained an ensemble of five NB/EM classifiers. We observed no significant improvement in classification accuracy with more than five classifiers trained.

The commonplace radial basis function kernel and sigmoid activation function were used for the SVM and NN models, respectively. The NN model was constructed with n-input neurons (one neuron for each attribute of the input vector), one hidden layer with $\frac{n}{2}$ neurons and a single output neuron to indicate boolean deal or no-deal. For J48, the Weka machine learning framework provided the implementation [Hall et al., 2009]. For SVM, the LibSVM library provided the functionality [Chang & Lin, 2011], while the Neural Net Framework (NNF, http://nnf.sourceforge.net) was used for the NN model.

We randomly selected a set of 2000 sentences from our industrial partner's database for positive (deal) samples. We complemented this set with 2000 negative (non-deal) samples from Project Guttenberg, also selected at random, to create a balanced 4000 sample training set. We transformed this sentence set into vectors using features from Table 2. Specifically, for our method NB/EM and J48, we constructed feature vectors in the form of <$\text{Word}_{[x]}$, ner_dateI, ner_organizationI,…,sym_SYM_posI> for every word [x] of the training set generating 80,000 individual training vectors with 12 attributes per vector. For SVM and NN, a sparse vector was constructed for each sentence, where each attribute corresponds to the frequency of a

recognized word followed by the remaining 11 features from Table 2; the result is a set of 4,000 individual training vectors with an average vector length of 1,811 tuples.

The model was trained and tested using ten-fold cross validation. The initial training set is divided into 10 equal-sized samples – nine for training and the tenth for testing. This process of training and testing is repeated 10 times with each of the samples becoming the testing set in one of the iterations. The average of these results is shown in Table 4 and includes a ± interval for a 95% confidence.

Table 4. Results of sentence classification for accuracy, sensitivity and specificity with 95% confidence interval (±).

%	NB/EM	J48	SVM/z-score	SVM/linear scaling	NN/z-score	NN/linear scaling
Accuracy	96 ± 0.4	88.7 ± 0.4	77.33 ± 0.2	51.39 ± 0.1	48.3 ± 0.3	48.3 ± 0.2
Sensitivity	91 ± 0.4	81.3 ± 0.4	78.1 ± 0.2	0	0	0
Specificity	99.7 ± 0.3	95.4 ± 0.4	77.5 ± 0.2	100	100	100

The NB/EM model performed the best followed by J48 and SVM/z-score. There was a noticeable improvement of SVM when trained with z-score normalized vectors versus linear scaling of the features. NN performed the worst regardless of which normalization method was used.

For completeness and further insight, Table 4 also includes sensitivity and specificity observations. The sensitivity statistic reveals how effectively the model can identify a deal sentence when it actually is a deal sentence. Specificity measures how well the model can classify a sentence as a non-deal sentence when the sentence it truly non-deal. Our NB/EM sensitivity results were noticeably better than J48 and SVM/z-score. For specificity, J48 was a close second. The sensitivity/specificity scores of 0/100 for SVM/linear scaling, NN/z-score, and NN/linear scaling suggest these models almost always label a sentence as non-deal.

These trained models were further tested. Ten rounds of samples of 600 sentences (300 deal/300 no-deal) were selected at random with replacement. Their individual classification accuracy was averaged with results given in Figure 8(a), while Figure 8(b) presents the resulting specificity vs. sensitivity plot. Once again, NB/EM performed the best with J48 a close second. SVM/z-score had a better than average accuracy whereas SVM/linear, NN/z-score and NN/linear struggled.

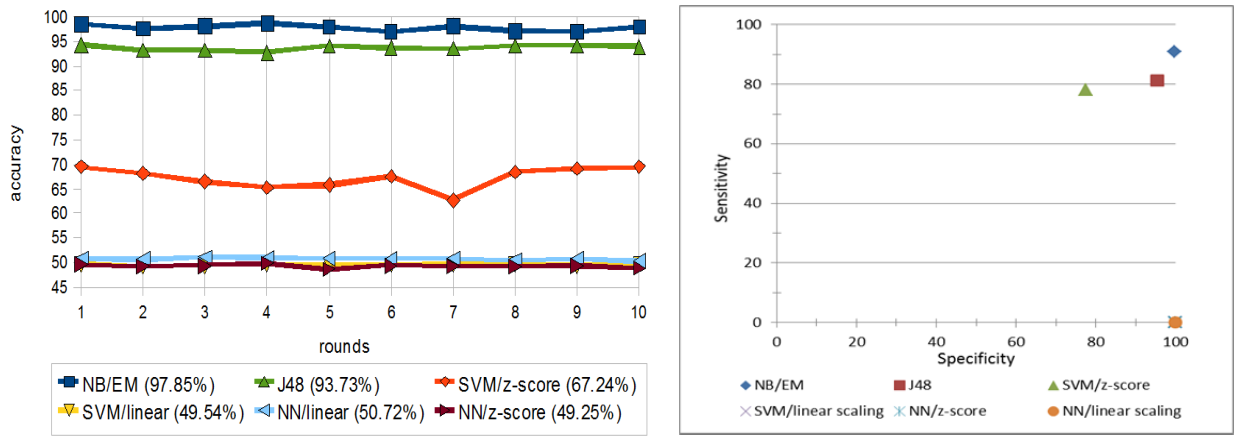


Figure 8. (a) Average classification accuracy for ten rounds of randomly sampled sentences. (b) Specificity versus Sensitivity plot.

In the next series of tests, the models were retrained with two different sets of features. The first set consisted of only the Words feature (A) while the second set contained all features except words (ALL–A). The words (A) versus the-rest (ALL–A) test was performed to determine the impact of this core attribute (Words). Results are given in Table 5.

Table 5. Accuracy, sensitivity and specificity using Words-only (A) versus the-rest (ALL-A). Average of 10 rounds of random sampling with 95% confidence interval (±).

	Word only feature (A)			**The rest (ALL-A)**		
%	**Accuracy**	**Sensitivity**	**Specificity**	**Accuracy**	**Sensitivity**	**Specificity**
NB/EM	97.52 ± 0.6	99.64 ± 0.6	95.82 ± 0.6	89.19 ± 0.6	98.71 ± 0.6	83.33 ± 0.6
J48	86.5 ± 0.5	81.15 ± 0.5	92.32 ± 0.5	88.27 ± 0.5	91.85 ± 0.5	88.30 ± 0.5
SVM/z-score	71.4 ± 1.0	69.74 ± 1.0	73.29 ± 1.0	89.05 ± 1.0	97.05 ± 1.0	84.10 ± 1.0
SVM/linear scaling	49.47 ± 0.2	0	100	49.5 ± 0.2	0	100
NN/z-score	48.60 ± 0.3	0	100	52.6 ± 0.3	0	100
NN/linear scaling	49.55 ± 0.2	0	100	49 ± 0.2	0	100

When trained only with the Words feature (A), the NB/EM model was equally effective as it was with all 12 features available. For the word-only model (A), our NB/EM performed best for both sensitivity and specificity, followed by J48 and SVM/z-score classifiers. For the non-words model (ALL-A), NB/EM performed best for sensitivity while J48 took top honors for specificity. NB/EM, J48, and SVM/z-score achieved similar accuracy when trained without the word feature (the-rest; ALL–A). This might suggest that the-rest attributes are unnecessary and thus the feature model can be reduced to the single feature. However, this is not the case as this suggestion presumes the test sentence always contains familiar words. These results demonstrate that the NB/EM model, relying on its other 11 features, can determine a sentence classification to 89% accuracy even when no recognizable words are present.

4.2 Evaluation of Web Page Classification Accuracy

In this section, we present the experiments we performed in order to evaluate the accuracy of our approach to classifying Web pages using the True/False ratio of labeled sentence blocks. It is at this step where our Deal Crawler decides on whether the visited page is one of interest (deal) by comparing the calculated True/False ratio against a predetermined threshold (equation 3). Through a series of three experiments, we highlight the challenges a crawler encounters when attempting to identify pages as either deal or non-deal. First, we present an experiment aimed at assessing the page-level accuracy of the proposed classifier, i.e., its ability to correctly classify an entire Web page as deal vs. no-deal. Second, we describe an experiment aimed at evaluating the classifier's ability to handle ambiguous terms. Third, we demonstrate the classifier's ability to re-classify search results as deal versus non-deal, and to re-rank those search results based on the presence of deals.

For this first experiment, we arbitrarily chose 42 Web pages that were clearly of our target deal domain (true). We counter-balanced these with an equally sized set of Web pages of non-deal status (false). This testing sample was not part of the corpus and consequently was never before seen by our model. For these 84 Web pages a true/false ratio was calculated using a true-false ratio (Γ) of 1.0 and the deal-sentence confidence threshold (τ) of 90%. Our justification for these default threshold values is tested in Section 4.5.

Table 6 gives a summary of the observations. The worst 6 true/false ratios from the deal group are showcased with the best 6 ratios of the no-deal class sorted from high to low. The table reveals two mislabelled pages with ratios 0.94 and 1.11 respectively and demonstrates the threshold value of 1.0 (Γ) is a suitable boundary for this classifier. This threshold value was a logical selection as it indicates that a Web page requires at least as many true blocks as it has false blocks.

Table 6. The first 6 rows present the worst true/false ratios obtained in the group of deal pages, while the next 6 rows show the best ratios taken from the no-deal group.

LABEL	(T)	(F)	(T/F)	SITE	
Deal	32	16	2	411travelbuys.ca	Worst scoring T/F from known deal pages (T/F≥1.0)
Deal	9	5	1.8	Dailycheckout.com	
Deal	22	13	1.69	Jewelrydeal.com	
Deal	17	13	1.31	Greendeals.org	
Deal	132	114	1.16	Dealtime.com	
Non-Deal	20	18	1.11	Filmsite.org	
Deal	16	17	0.94	Thomascook.ca	Best scoring T/F from known non-deal pages (T/F<1.0)
Non-Deal	11	13	0.85	Rome.info	
Non-Deal	24	31	0.77	Lvoe.ca	
Non-Deal	5	7	0.714	Rockyview.ab.ca	
Non-Deal	20	28	0.71	Foodnetwork.com	
Non-Deal	22	33	0.67	Howto.wired.com	

For the second experiment, we demonstrate how our algorithm can re-evaluate search results from general purpose search engines (Google/Bing) towards favouring our target domain. We begin with a simple Google search using only the single term “soap”. Of the top-50 ranked results, only 12 linked to a Web page of interest to our target domain of daily deals. Table 7 lists some of particularly interesting results that demonstrate the seemingly ubiquitous use of the term soap, especially as an acronym.

Table 7. Partial top-50 Google search results for the query “soap”.

Olay® Bar Soap
Simple Object Access Protocol(SOAP) 1.1
SOAPCREATIVE - DIGITAL CREATIVE AGENCY
SOAP- Wikipedia
Soap (TV Series 1977–1981) - IMDb
SoapOpera News and Updates at Soaps.com
SOAP- Society for Obstetric Anesthesia and Perinatology
Apache SOAP- the Apache Web Services™ Project
PHP: SOAP- Manual
SOAP PEAR
SOAP:: Short Oligonucleotide Analysis Package
Project SOAP - The Study of Open Access Publishing
SOAP::Lite for Perl
SOAP2012 - Sable Research Group - McGill University

We labeled these 50 links using the NB/EM classifier. The results are given in Table 8. The table shows no false positives with 100% precision and 67% recall. The strategy of using true/false ratio has the added benefit of providing a ranking of evaluated pages; with larger ratios indicating the potential for a high concentration of product offerings relative to all the other extraneous content within the page.

Table 8. Confusion matrix for “soap” Web page classification.

	ACTUAL TRUE	**ACTUAL FALSE**	**TOTAL**
NB/EM TRUE	8 True Positive	0 False Positive	8
NB/EM FALSE	4 False Negative	38 True Negative	42
TOTAL	12	38	50

Table 9 serves to demonstrate the potential for our approach as a ranking method for the set of visited Web pages. The ideal model is one which gives ratios approaching infinity for those pages that are actual deals (false → 0) while assigning a ratio of zero to actual non-deal pages (true → 0).

Table 9. Web pages with highest True/False ratios.

Web Page	True (T)	False (F)	T/F
Upper Canada Soap	4	1	4
Soap.com: Health & Beauty Products	11	3	3.6667
Soap & More	9	4	2.25
Handmade Soap – Rocky Mountain Soap	40	20	2
Olay Soap	7	4	1.75
ThinkGeek: Shower Shock Caffeinated Soap	32	19	1.6842
Olay Bar Soap	6	4	1.5
Natural Soap – Vegitable Soap from Mountain Ski	9	6	1.5
ThinkGeek: Bacon Soap	22	20	1.1
Soap – Lush	24	22	1.0909
Welcom to Canwax Inc – Candle & Soap	10	10	1
What is SOAP? – Webopedia	23	27	0.8519

In our third experiment we expand the above "soap" test by selecting 10 product terms frequently seen in deal Websites such as GROUPON. These terms were Googled and their top-40 ordinal rank positions observed. The top-40 was re-ranked by the true-false (T/F) ratio, similar as in the "soap" experiment. The recall of deal-like pages computed before and after re-ranking of the top-10, top-20, top-30 and top-40 results, is shown in Table 10.

Table 10. The recall measure computed before and after re-ranking of top-10, top-20, top-30 and top-40 ordinal rankings.

	Google Original Ranking				True/False Ratio Re-ranking			
	Top-10	**Top-20**	**Top-30**	**Top-40**	**Top-10**	**Top-20**	**Top-30**	**Top-40**
Indoor tanning	$0.0_{(0)}$	$0.50_{(2)}$	$0.75_{(3)}$	$1.0_{(4)}$	$0.75_{(3)}$	$0.75_{(3)}$	$0.75_{(3)}$	$1.0_{(4)}$
Laser eye correction	$0.24_{(6)}$	$0.40_{(10)}$	$0.72_{(18)}$	$1.0_{(25)}$	$0.24_{(6)}$	$0.48_{(12)}$	$0.72_{(18)}$	$1.0_{(25)}$
Facials	$0.24_{(7)}$	$0.52_{(15)}$	$0.69_{(20)}$	$1.0_{(29)}$	$0.31_{(9)}$	$0.59_{(17)}$	$0.79_{(23)}$	$1.0_{(29)}$
Canada travel	$0.20_{(2)}$	$0.30_{(3)}$	$0.70_{(7)}$	$1.0_{(10)}$	$0.50_{(5)}$	$0.80_{(8)}$	$0.90_{(9)}$	$1.0_{(10)}$
Daycares	$0.20_{(3)}$	$0.40_{(6)}$	$0.60_{(9)}$	$1.0_{(15)}$	$0.47_{(7)}$	$0.80_{(12)}$	$0.93_{(14)}$	$1.0_{(15)}$
Italian food	$0.22_{(5)}$	$0.48_{(11)}$	$0.78_{(18)}$	$1.0_{(23)}$	$0.30_{(7)}$	$0.61_{(14)}$	$0.83_{(19)}$	$1.0_{(23)}$
Acupuncture treatments	$0.17_{(2)}$	$0.25_{(3)}$	$0.58_{(7)}$	$1.0_{(12)}$	$0.17_{(2)}$	$0.50_{(6)}$	$0.75_{(9)}$	$1.0_{(12)}$
Christmas trees	$0.24_{(6)}$	$0.48_{(12)}$	$0.72_{(18)}$	$1.0_{(25)}$	$0.36_{(9)}$	$0.72_{(18)}$	$0.96_{(9)}$	$1.0_{(25)}$
Dry cleaning	$0.19_{(5)}$	$0.44_{(12)}$	$0.74_{(20)}$	$1.0_{(27)}$	$0.33_{(9)}$	$0.67_{(18)}$	$0.89_{(24)}$	$1.0_{(27)}$
Smartphones	$0.26_{(7)}$	$0.56_{(15)}$	$0.74_{(20)}$	$1.0_{(27)}$	$0.37_{(10)}$	$0.67_{(18)}$	$0.81_{(22)}$	$1.0_{(27)}$

Table 10 demonstrates how general-purpose search engines do not necessarily return pages that are offering a product or a service even when there is no ambiguity in the terms. For example, with the term "indoor tanning", only four results were those of our target domain (deal). The others included pages on the health controversy of indoor tanning, the skin cancer, and legislation limiting indoor tanning use by minors. Consequently, the original ordinal ranking by Google listed none of our target domain four pages in the top-10. After re-ranking by the T/F ratio, three of the four target domain links appeared in the top-10 (recall 0.75). Figure 9 illustrates the improvement in recall based on our approach, i.e., after T/F re-sorting of the 10 sample phrases for top-10, top-20, and top-30 ordinal positions. We exclude the top-40 in this graph since the recall is always 1.0 as shown in Table 10.

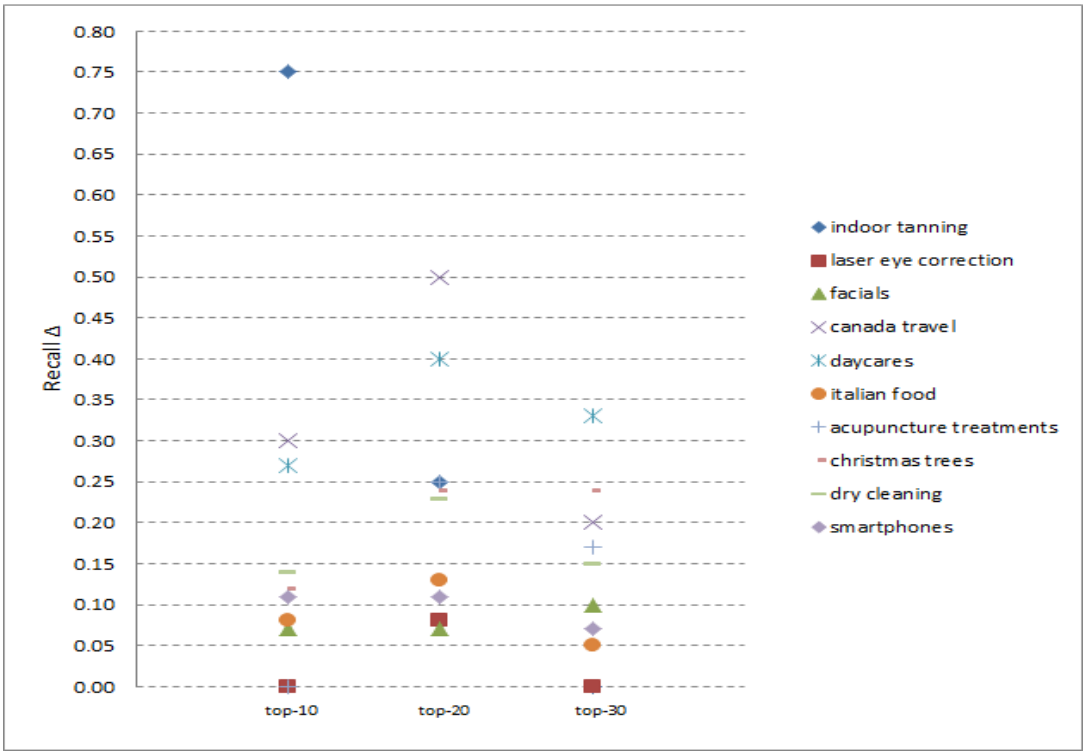


Figure 9. Recall improvement (Δ = after-before) for top-10, top-20, and top-30 for sampled terms.

4.3 Evaluation of the Segmentation and Localization Algorithms

The experimental set for this evaluation came from Section 4.2 which comprises 42 individual deal Web pages each from a different Web site. These 42 positive deal Web pages accounted for 1,402 individual product offerings identified by human reviewers.

The criteria used for determining whether the segmentation/localization process was successful were as follows:

Criteria 1: A block is correctly localized as a deal if and only if the block makes reference to exactly one product offering. If the block contains information for more than a single deal then it was under-partitioned and should have undergone further segmentation in order to have its contents split into individual deals.

Criteria 2: Because the descriptiveness of a deal will vary significantly across Web sites, the minimum amount of information necessary for a product offering is the name of the product and its price. Blocks that do not meet this minimum are considered to be over-partitioned and an indication of the failure of either the sanity check stopping criteria or a failure of the tree pruning algorithm (Section 3.3).

Criteria 3: A block that satisfies criteria 1 and 2 but makes reference to the same deal as one or more siblings will only get credit for correctly classifying the deal once.

We proposed the use of the general heuristic rules of section 3.3 to aid in meeting this criteria.

We summarize the results in Table 11 sorted by descending F-score. Based on the evaluation criteria outlined above, the experiment demonstrated favorable results with the overall F-score of 0.903. The algorithm correctly identified 1,282 deals with 154 misclassifications (false positives). Not surprisingly, the best results were with sites whose Web pages appeared to be dynamically generated through a local database that applied a consistent template layout. Since our segmentation algorithm (see Algorithm 1) relies on long frequent patterns of HTML sequences, correctly identifying the template pattern tend to yield high performance.

Table 11. Deal localization statistics for 42 parsed Web pages.

Site	Actual Deals	AI Deals	Right	Wrong	Prec.	Recall	F-score
Dailynews.com	8	8	8	0	1.000	1.000	1.000
Trackdailydeals.com	13	13	13	0	1.000	1.000	1.000
Deals.com	25	25	25	0	1.000	1.000	1.000
Dealextreme.com	52	53	52	1	0.981	1.000	0.990
Mydealbag.com	246	238	238	0	1.000	0.967	0.983
	↑	**TOP 5**		↑			
(32 sites aggregated)	950	970	869	101	0.896	0.915	0.905
	↓	**BOTTOM 5**		↓			
Sidebuy.com	9	17	9	8	0.529	1.000	0.692
Music123.com	20	13	11	2	0.846	0.550	0.667
Dealfrenzy.com	5	10	5	5	0.500	1.000	0.667
Elivedeals.com	53	64	39	25	0.609	0.736	0.667
Rubywallet.com	21	25	13	12	0.520	0.619	0.565
Totals/Averages:	**1402**	**1436**	**1282**	**154**	**0.893**	**0.914**	**0.903**

4.4 Feature Analysis

We supplement the experiments of section 4.1, with feature sensitivity analysis on the-rest attributes (i.e., all the features except the Words feature) to determine the importance of individual features and to identify opportunities for a reduced set of attributes. Recall that for notational simplicity, the features given in Table 2 are labeled from A to L, and the overall set of features is referred to as ALL.

We utilized a common technique of recursive feature elimination in which a model is retrained and tested with one less feature from its current set [Li & Yang, 2005]. Beginning with the current 'best' base set of features defined as the-rest (ALL–A), we generate 11 new models from the random set of 2000 positive and 2000 negative samples, each having one feature less than the base, and re-test these models using ten rounds of random samples of size 300 positive and 300 negative sentences. Table 12 gives the results.

The results indicate that the feature ner_moneyl (G) has the greatest impact on accuracy. Recall, this feature is the count of the tags identified by named entity recognizer as being of type currency. Interestingly, the feature sym_DollarAvg (I), which is the average dollar amount of the currency tags, does not seem to influence the model. Together, these observations suggest that

any dollar amount is significant to the model over the actual price of any particular product. The feature ner_dateI (B) also has certain level of impact, as its removal results in a 4% drop in accuracy. This is expected since product promotions are often conditional on the date they begin and expire. Finally, the model that omits sym_SYM_posI (ALL-A-L) improves the classification accuracy slightly from the base of 89.19% to 90.13%.

Table 12 also includes Pearson correlation tests between the base model (ALL-A) and the other reduced-by-one models. Models with Pearson's correlation close to zero indicate that the models behave independently; values close to +1 indicate the models are similar in behavior, while values approaching -1 indicate they behave inversely to each other. For example, the model ALL-A-K has a Pearson coefficient of 0.96 indicating that the removal of the feature sym_CD_posI (K) from the-rest base model (ALL-A), results in a model that is very similar to its base.

Table 12. Recursive feature elimination for the-rest (ALL–A) model.

Pearson	**NB/EM Model**	**%**	∆
	Base: the-rest (ALL-A)	**89.19**	
0.7	Ner_dateI(ALL-A-B) *Ex: January 1, 2013*	84.98	-4.21
0.87	Ner_organizationI(ALL-A-C) *Ex:Ryerson University*	89.66	0.47
0.98	Ner_timeI(ALL-A-D) *Ex:12:45 pm*	89.20	0.01
0.89	Ner_locationI(ALL-A-E) *Ex: Vancouver, BC, Canada*	89.30	0.11
0.95	Ner_percentageI(ALL-A-F) *Ex:100%*	89.64	0.45
0.14	Ner_moneyI(ALL-A-G) *Ex: $1000*	68.12	-21.07
0.86	Ner_personI(ALL-A-H) *Ex:John Smith*	89.38	0.19
0.86	Sym_DollarAvgI(ALL-A-I) *Ex:42.13 (avg $)*	88.92	-0.27
0.72	Sym_PercentAvgI(ALL-A-J) *Ex:50.25 (avg %)*	89.76	0.57
0.96	Sym_CD_posI(ALL-A-K) *Ex: 5 (any number)*	89.62	0.73
0.79	Sym_SYM_posI(ALL-A-L) *Ex:!, #, ?, +*	90.13	0.94

We repeat the elimination using the new local maximum of (ALL-A-L) as the base. At the end of the iteration, the performance of the reduced-by-one models is examined for an improvement from the base. The next iteration of eliminations occurs if a better or equal reduced model is found. Algorithm 2 outlines the elimination procedure.

Input: A trained NB/EM model β

Output: A feature reduced model of equal or better accuracy than β

Algorithm:

1. Begin with base NB/EM model (β) with all the features.
2. Compute base model accuracy (δ).
3. Let $\eta_{best} \leftarrow$ null
4. Let f be the set of features of β
5. FOR feature υ in f DO
   5.1 Train new model (η) without feature υ. η←β \ υ
   5.2 Compute accuracy (t) for model η.
   5.3 IF t ≥ δ THEN
      5.3.1 Let δ ← t
      5.3.2 Let $\eta_{best} \leftarrow \eta$
   5.4 END IF
6. END FOR /* NEXT feature υ */
7. IF η_{best} NOT null THEN
   7.1 β←η_{best}
   7.2 goto step 3
8. END IF

Algorithm 2. The search for the best feature reduced NB/EM model.

The most accurate model reduced from the-rest base was observed when features sym_SYM_posI (L) and ner_timeI (D) were removed giving the 90.27% accuracy; an improvement of 1.08% from the-rest base. Although this improvement is nominal, the difference translates to an additional 11 mis-classifications per 1000 sentences and consequently may skew the true/false ratio of deal versus no-deal sentences used to determine the overall label of a Web page. Future follow-up experimentation involves a comparison of results presented in tables 6, 8, 9, 10, and 11 with the results for the feature-reduced model (ALL-A-L-D) to quantitatively observe the effects of the 1.08% differential.

4.5 Parameter Settings

Recall that the proposed classification method involves two threshold values: the minimum probability a sentence block must achieve for a deal label (τ), and the minimum true/false block ratio an entire Web page has to have to be classified as a deal (Γ). Our threshold values are 0.90 and 1.0, respectively. They have been chosen based on observations of numerous test runs. In this section, we examine the appropriateness of these selections.

To test the minimum probability sentence block threshold (τ), we selected 29,332 sentences with balanced deal/no-deal distribution at random from our corpus (section 3.1). We then computed the classification accuracy for this set using thresholds from 1 to 100 at 10 percent intervals. The results are presented on Figure 10 showing peaks of 94.5% for threshold values of 80 and 90 percent.

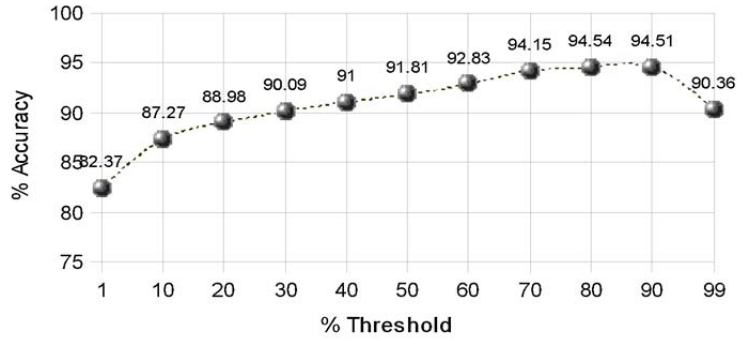


Figure 10. Classification accuracy at various thresholds for the minimum probability a sentence block must achieve to be labeled as a deal

Furthermore, we revisited the experiments presented in section 4.2 with differing True/False ratios (Γ). We observed a peak F-score of 80% at 1.0 True/False ratio (Figure 11). The presented results provide empirical support for the chosen value for both thresholds.

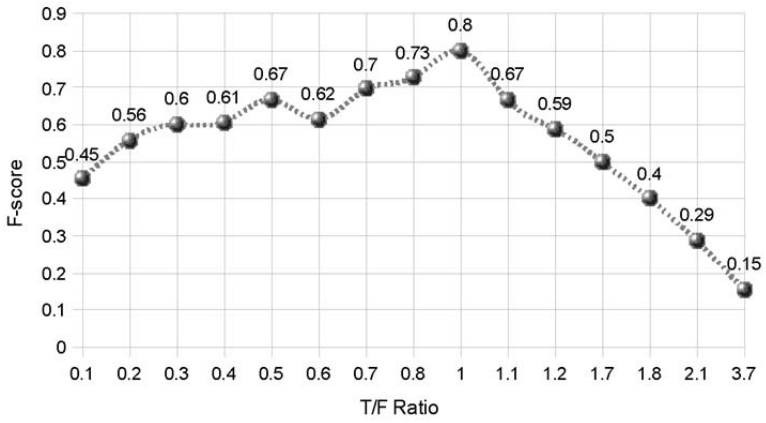


Figure 11. F-score for the Web page classification accuracy experiment with different True/False ratios

5. Discussion

Our solution revolves around a Naïve Bayes (NB) classifier with an Expectation/Maximization (EM) clusterer. This pairing of NB with EM is common [Calders & Verwer, 2010] and offers advantages to using NB alone. NB classifiers often perform well with text [Kim et al., 2003]. They are relatively easy to deploy and are fast classifiers. Speed is an important consideration in our model of the intelligent deal crawler, which must quickly scour the Internet for products whose availability and pricing can change frequently. The probability calculation of equations 2 and 3 can efficiently be performed once the model has been trained. This speed and ease of the NB approach is offset by the independence of features assumption it relies on. However, by combining NB with EM, which can account for feature dependency, we can construct acceptable deal and non-deal clusters. The independence condition of NB is further lessened as its task is simply to predict the most likely cluster the input sentences belong to.

EM can also compensates for an unbalanced training set of positive and negative examples [Tsuruoka & Tsujii, 2003], and allows for incomplete training samples with missing or unknown attributes to be included in the analysis. Acquiring the positive samples for training was straightforward thanks to the deal database provided by our industrial partner. Conversely, sufficient numbers of negative samples for training is sometimes a challenge, but was not the case with our classifier since it operates at the sentence level. Yu et al. [2002] tackled the problem of negative training samples with an SVM classifier that used labeled positive examples intermixed with unlabeled pages. In our method, negative samples were easily gathered from Project Guttenberg containing thousands of sentences suitable as counter-examples. From these samples, the features of Table 2 were chosen by the need for efficient computation due to the deal crawler's functional requirement of quick identification of a possible deal source. Thus, deeper analysis techniques such as semantic role labeling were avoided due to their time and computational overhead, although a limited use of those techniques in the manner suggested in [Ciaramita et al., 2008] is planned for the future work. Furthermore, features derived from the visual appearance of a Web Page such as the location of HTML tags and the colors used in the page were avoided given the heterogeneity of the visual styles used in different daily deal websites. Our experience showed that not only the visual representation styles of daily deal websites are mutually different, but also a single daily deal website could undergo dramatic visual representation changes in short periods of time. Moreover, the inclusion of such features would prevent us from using resources such as Project Guttenberg as a negative sample training source.

In section 4, we conducted various experiments to test the effectiveness of our model's binary classification, page segmentation, and deal localization abilities. Utilizing equation 2 with $\tau = 0.90$, we observed our sentence-level classification accuracy (Section 4.1) was highest with our NB/EM model followed by Decision Tree (J48) then Support Vector Machine (SVM) and Neural Network (NN) using the same Table 2 features.

We also observed a significant disparity in accuracy between NB/EM and J48 versus SVM and NN models which may be attributed to numerous conditions. The structure of a sample vector for SVM and NN models are significantly different from NB/EM and J48. Specifically, SVM and NN use a sparse vector of attributes where each attribute position corresponds to a recognized word and its frequency count plus additional 11 attributes for the-rest features. This results in a large vector of attributes based on the words encountered during training. For example, for the sampling of 4,000 sentences, the vector averaged 1,811 attributes (1,800 unique words plus 11 static the-rest attributes). In contrast, NB/EM and J48 models can represent the Words feature in a single attribute thus having a simpler model representation of a fixed 12-attribute vector. Although this sparseness does not necessarily represent a problem, particularly for SVMs where sparse feature vectors are commonly used, this situation presents a few considerations of its own. First, the size of the training set may need to be larger in order to produce sufficient unique vectors to adequately train the model. SVMs operate by finding a maximal separating hyperplane across multiple dimensions. A 1,811-attribute vector requires a separation plane for 1,811 dimensions, thus potentially requiring a larger training set to achieve a well-represented separation. Second, the importance of vector normalization is well-known in such ML models, hence the significant impact observed in accuracy with the change of normalization methods:

linear versus z-score. These considerations appear to have been realized in Table 5 with the removal of the Words feature. In this test, the vector attribute length shrunk from 1,811 down to a fixed 11 (the-rest) - resulting in identical vector of attributes for NB/EM, J48 and SVM. This reduction, combined with z-score normalization, gave SVM the same level of accuracy as NB/EM and J48. Comparatively, the NN model may benefit from a combination of different selection of parameters such as a change in activation function, number of hidden layers, number of neurons per layer, adjusted learning rate as well as a different normalization method and larger sample training size. Further investigation is needed but the number of model parameter adjustments necessary make this model difficult to tweak.

For assessing the Web Page classification capability (Section 4.2), our first experiment examined the accuracy of a deal or non-deal determination using equation 3 and an arbitrarily chosen tunable threshold Γ. The experiment concluded that a threshold $\Gamma = 1.0$ performed well with only two misclassifications of 84 Web Pages tested. In the second experiment, we googled the ambiguous term “soap” and used our method successfully as a filter to remove suggestions that were not of our target product domain and demonstrated how the value of equation 3 can be used as a measure of the strength of the deal/no-deal classification decision. In the third experiment, we rank the computed ratio of a deal classification for a set of Web Page candidates. Instead of a binary true/false determination, we ignore the threshold Γ and alternatively sort the value of equation 3 from high to low thus demonstrating the re-ordering potential of our algorithm.

In section 4.3 we test our method’s ability to segment and localize regions of a Web Page into deal and non-deal areas. Our segmentation technique involves directing the Web page classifier to bounded areas of a Web Page via recursive division; a technique also utilized by Cao et al. [2010] in what they described as their “iterative shrinking and dividing” strategy. These bounded areas are defined by the longest frequent patterns (LFPs) of HTML sequences within each region. Hwang et al. [2008] also searched for LFPs in an attempt to transform Web Pages into a mobile device friendly layout. Instead of LFPs, Mahmud et al. [2007] employed SVM to identify segments of “importance” based on the context of a surrounding link. Our deal localization strategy involves pruning through a set of heuristic rules (Section 3.3) to ensure the identified region satisfies the requirement of one deal per region. Yu et al. [2003] merged heuristic rules with VIPS for block extraction and boundary detection in order to discard sections of the Web page with little or no semantic substance, while Yin and Lee [2005] constructed a graph model that labels regions into one of five explicit categories.

In section 4.4 we investigate the effectiveness of the features of Table 2 and propose a feature-reduce-by-one method (Algorithm 2) to simplify the model and improve accuracy. Hsu & Lu [2008] used an increase-by-one selection method also based on Pearson Correlation Coefficient and a distance function, while Selamat et al. [2003] used Principal Component Analysis (PCA) for reducing the feature set. It can be shown that the number of models to train using Algorithm 2 is bounded by the equation:

$$\sum_{i=1}^{n-1} \frac{n(n-1)}{2} \qquad (4)$$

where n is the number of features of the base model. In our circumstance, this gives 66 models which is a substantial reduction from the total search space of 4,095 models we would obtain by combining in all possible ways the 12 features presented in Table 2. If this reduced set is still too large, Pearson's coefficient may be used to selectively limit the features in the next iteration. Namely, only those features that exceed a determined Pearson's threshold would advance to the next iteration. It is also important to mention that this algorithm is a greedy hill-climbing algorithm that may converge to a local maxima.

Our proposed model depends on two tunable parameters: the deal sentence threshold τ for equation 2 and the $\frac{deal}{no-deal}$ ratio threshold Γ of equation 3. For completeness we conclude our tests by empirically justifying our chosen threshold defaults of $\tau = 0.90$ and $\Gamma = 1.0$ (Section 4.5). We end our discussion with Table 13 summarizing our classification, segmentation, and localization methods alongside similar methods in literature.

Table 13. A summary of various classification, segmentation and feature analysis methods.

Authors	**Classification Method**	**Segmentation Method**	**Feature Analysis**
Jcuzzola et al.	NB/EM model examining natural language at sentence level.	Recursive application of classification model on LFP with heuristic rules.	Reduce-by-one using PCC.
Selamat et al. [2003]	Multiple category classification for sports articles with neural net		Principal Component Analysis (PCA).
Hsu & Lu. [2008]	SVM to search for disease-causing genes.		Increase-by-one with Euclidean Distance and PCC.
Yu et al. [2002]	SVM that used labeled positive examples intermixed with unlabeled pages.		
Fiol-Roig et al. [2011]	Multiple variants of decision trees.		
Mahmud et al. [2007]		SVM to identify segments based on the context of a surrounding link.	
Kohlschutter and Nejdl [2008]		Text-based relying on a text density metric.	
Hwang et al. [2008]		Segmentation using LFP pattern matching.	
Cao et al. [2010]		Recursive division. Iterative shrinking and dividing.	
Yu et al. [2003]		Heuristic rules with VIP.	
Yin and Lee [2005]		Graph model that labels regions into categories.	

6. Conclusion

In this paper, we have presented our research related to the development and evaluation of a unique framework for the classification of Web pages based on the presence of daily deal information, and for localizing such information within each page classified as a deal page. The presented methods function in a semi-supervised manner, without prior knowledge of the structure or the content of the Web pages being considered. Therefore, the proposed solution is capable of reducing the heavy reliance on human intervention for finding daily deals, which is one of the greatest challenges faced by the current deal aggregators.

Our future work will involve extracting structured property value pair data from the deal-identified regions. We will leverage well known ontologies and knowledge bases from specific subject domains, available on the Web as Linked Open Data[3], to add semantic meta-data to localized deals. We will further incorporate natural language processing to identify entities such as persons, locations, organizations and similar, and link them to the chosen ontologies. We are developing disambiguation algorithms to match these entities to the most-likely ontology concepts; then through pattern-matching and statistical analysis identify property/value attributes suitable for the product identified. The goal is to draw correspondences between the textual descriptions and the concept properties in such a way that the values for concept properties can be derived from the localized text.

References

[Ahmadi et al, 2011] Ahmadi, A., Fotouhi, M., and Khaleghi, M. 2011. Intelligent classification of Web pages using contextual and visual features. Appl. Soft Comput. 11, 2 (March 2011), 1638-1647.

[Cai et al., 2003] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. 2003. VIPS: a vision-based page segmentation algorithm. Technical Report MSR-TR-2003-79. Available at: http://research.microsoft.com/pubs/70027/tr-2003-79.pdf

[Calders & Verwer, 2010] Calders, T.G.K. & Verwer, S.E. (2010). Three naive Bayes approaches for discrimination-free classification. Data Mining and Knowledge Discovery, 21(2), 277-292.

[Cao et al., 2010] Cao, J., Mao, B., & Luo, J. 2010. A segmentation method for Web page analysis using shrinking and dividing, International Journal of Parallel, Emergent and Distributed Systems, 25:2, 93-104.

[Chakrabarti et al., 2008] Chakrabarti, D., Kumar, R., and Punera, K. 2008. A graph-theoretic approach to Web page segmentation. In *Proceedings of the 17th international conference on World Wide Web* (WWW '08). ACM, New York, NY, USA, 377-386.

[Chang & Lin, 2011] C. C. Chang and C. J. Lin. (2011) LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3), 27:1--27:27.

[3] http://datahub.io/group/about/lodcloud

[Chau & Chen, 2008] Chau, M., & Chen, H. 2008. A machine learning approach to Web page filtering using content and structure analysis. *Decis. Support Syst.* 44, 2 (January 2008), 482-494.

[Chen & Hsieh, 2006] Chen, R. C., & Hsieh, C. H. 2006. Web Page Classification Based On A Support Vector Machine Using A Weighted Vote Schema. Expert Systems with Applications, vol. 31, pp. 427-435.

[Ciaramita et al., 2008] Ciaramita, M., Attardi, G., Dell'Orletta, F., Surdeanu, M. (2008). DeSRL: a linear-time semantic role labeling system. In Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL '08). Association for Computational Linguistics, Stroudsburg, PA, USA, 258-262.

[Debnath et al., 2005] Debnath, S., Mitra, P., Pal, P., & Lee Giles, C. 2005. Automatic Identification of Informative Sections of Web Pages. IEEE Trans. on Knowl. and Data Eng. 17, 9 (September 2005), 1233-1246.

[Dietterich, 2000] Dietterich, T.G. 2000. Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems* (MCS '00), Josef Kittler and Fabio Roli (Eds.). Springer-Verlag, London, UK, UK, 1-15.

[Fiol-Roig et al., 2011] Fiol-Roig, G., Miró-Julià, M., Herraiz, E. (2011). Data Mining Techniques for Web Page Classification. PAAMS (Special Sessions) 2011: 61-68

[Ghigliotty, 2011] Ghigliotty, D. (2011) Do You Really Want a Job at Groupon? The Wall Street Journal, November 2, 2011. Retrieved from: http://online.wsj.com/article/SBB10001424052970204528204577012073472414832.html

[Hall et al., 2009] Hall, M. Eibe, F., Holmes, G. Pfahringer, B., Reutemann, P. Witten, I. (2009) The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1.

[Hao et al., 2011] Q. Hao, R. Cai, Y. Pang, and L. Z. Zhang. From One Tree to a Forest: a Unified Solution for Structured Web Data Extraction. In Proceedings of SIGIR '11, Beijing, China, 2011. ACM Press, pp. 775–784.

[Hsu & Lu, 2008] Hsu, H., Lu, M. (2008). Feature Selection for Cancer Classification on Microarray Expression Data, Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on , vol.3, no., pp.153-158, 26-28.

[Hwang et al., 2003] Hwang, Y, Kim, J., & Seo, E. 2003. Structure-aware web transcoding for mobile devices. IEEE Internet Computing, 7(5),14–21.

[Kang et al., 2010] J. Kang, J. Yang, J. Choi. 2010. Repetition-based Web Page Segmentation by Detecting Tag Patterns for Small-Screen Devices, IEEE Transactions on Consumer Electronics, vol. 56, no. 2, pp. 980-986.

[Kim et al., 2003] Kim, S., Seo, H., Rim, H. (2003). Poisson naive Bayes for text classification with feature weighting. In Proceedings of the sixth international workshop on Information retrieval with Asian languages (AsianIR '03), Vol. 11. Association for Computational Linguistics, Stroudsburg, PA, USA, 33-40.

[Kohlschütter & Nejdl, 2008] Kohlschütter, C., & Nejdl, W. 2008. A densitometric approach to Web page segmentation. In *Proceedings of the 17th ACM conference on Information and knowledge management* (CIKM '08). ACM, New York, NY, USA, 1173-1182.

[Li & Yang, 2005] Li, F., Yang, Y. (2005). Analysis of recursive feature elimination methods. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '05). ACM, New York, NY, USA, 633-634.

[Lin & Ho, 2002] S. Lin and J. Ho. 2002. Discovering informative content blocks from Web documents. Proc. 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, pp. 588-593.

[Liu et al., 2004] Liu, B., Grossman, R., & Zhai, Y. 2004. Mining Web Pages for Data Records. IEEE Intelligent Systems, pp. 49-55.

[Mahmud et al., 2007] Jalal U. Mahmud, Yevgen Borodin, and I. V. Ramakrishnan. 2007. Csurf: a context-driven non-visual web-browser. In Proceedings of the 16th international conference on World Wide Web, WWW '07, pages 31–40, New York, NY, USA, ACM.

[Miao et al., 2009] Miao, G., Tatemura, J., Hsiung, W.P., Sawires, A., & Moser, L. E. 2009. Extracting data records from the web using tag path clustering. In Proceedings of the 18th international conference on World wide web (WWW '09). ACM, New York, NY, USA, 981-990.

[Mulpuru et al., 2013] Mulpuru, S., Johnson, C., Roberge, D. 2013. Forrester: US Online Retail Sales To Reach $370 Billion By 2017, March 13, 2013. Retrieved from: http://www.forrester.com/US+Online+Retail+Forecast+2012+To+2017/fulltext/-/E-RES93281?objectid=RES93281

[Ozel, 2011] Özel, S.A. 2011. A Web page classification system based on a genetic algorithm using tagged-terms as features. Expert Systems with Application, Vol.38, pp. 3407-3415.

[Qi and Davidson, 2009] Qi, X., & Davison, B.D. 2009. Web page classification: Features and algorithms. ACM Comput. Surv. 41, 2, Article 12 (February 2009).

[Selamat et al., 2003] Selamat, A., Omatu, S., Yanagimoto, H., Fujinaka, T., Yoshioka, M. (2003) Web page classification method using neural networks. IEEJ Transactions on Electronics, Information and Systems, 123 (5). pp. 1020-1026.

[Song et al., 2012] Song, D., Wu, Y., Liao, L., Li, L., & Sun, F. 2012. A dynamic learning framework to thoroughly extract structured data from Web pages without human efforts. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (MDS '12). ACM, New York, NY, USA.

[Tsuruoka & Tsujii, 2003] Tsuruoka, Y., & Tsujii, J. (2003). Training a naive bayes classifier via the EM algorithm with a class distribution constraint. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 (CONLL '03), Vol. 4. Association for Computational Linguistics, Stroudsburg, PA, USA, 127-134.

[Vineel, 2009] Vineel, G. 2009. Web page DOM node characterization and its application to page segmentation. In *Proceedings of the 3rd IEEE international conference on Internet multimedia services architecture and applications* (IMSAA'09). IEEE Press, Piscataway, NJ, USA, 325-330.

[Wu et al., 2009] Wu, B., Cheng, X., Wang, Y., Guo, Y., and Song, L. 2009. Simultaneous Product Attribute Name and Value Extraction from Web Pages. In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03(WI-IAT '09), Vol. 3. IEEE Computer Society, Washington, DC, USA, 295-298.

[Yesilada, 2011] Yesilada, Y. 2011. Web Page Segmentation: A Review. University of Manchester and Middle East Technical University Northern Cyprus Campus.

[Yin et al., 2005] Yin, X., & Lee, W. S. 2005. Understanding the function of web elements for mobile content delivery using random walk models. In Special interest tracks and posters of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, DOI:10.1145/1062745.1062913

[Yu et al., 2002] Yu, H., Han, J. Chang, K. (2002). PEBL: positive example based learning for Web page classification using SVM. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02). ACM, New York, NY, USA, 239-248.

[Yu et al., 2003] Yu, S., Cai, D., Wen, J.R., & Ma, W.Y. 2003. Improving pseudo-relevance feedback in web information retrieval using Web page segmentation. In Proceedings of the 12th international conference on World Wide Web, WWW ’03, pages 11–18, New York, NY, USA. ACM